IET Computers & Digital Techniques

Research Article

IET Journals
The Institution of Engineering and Technology

Comparative analysis of network-on-chip simulation tools

ISSN 1751-8601
Received on 8th April 2017
Revised 5th August 2017
Accepted on 12th September 2017
E-First on 5th December 2017
doi: 10.1049/iet-cdt.2017.0068
www.ietdl.org

Sarzamin Khan[1] ✉, Sheraz Anjum[2], Usman Ali Gulzari[3], Frank Sill Torres[4]

[1]Department of Electrical Engineering, COMSATS Institute of Information Technology, Wah Cantt, Pakistan
[2]Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt, Pakistan
[3]Department of Electrical Engineering, COMSATS Institute of Information Technology, Islamabad, Pakistan
[4]Department of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil
✉ *E-mail: khan.szamin@yahoo.com*

Abstract: Network-on-chip (NoC) is a reliable and scalable communication paradigm deemed as an alternative to classic bus systems in modern systems-on-chip designs. Consequently, one can observe extensive multidimensional research related to the design and implementation of NoC-based systems. A basic requirement for most of these activities is the availability of NoC simulators that enable the study and comparison of different technologies. This study targets the analysis of different NoC simulators and highlights its contributions towards NoC research. Various NoC tools such as NoCTweak, Noxim, Nirgam, Nostrum, BookSim, WormSim, NOCMAP and ORION are evaluated and their strengths and weaknesses are highlighted. The comparative analysis includes methods for estimation of latency, throughput and energy consumption. Further, the exemplary real world application, video object plane decoder is mapped on a 2D mesh NoC using different mapping algorithms under NOCMAP and NoCTweak simulators for comparative analysis of the NoC simulators and their embedded mapping algorithms.

1 Introduction

Modern integrated system designs incorporate the system-on-chip (SoC) technology in order to improve performance, costs and energy consumption. Embedded SoC systems are comprised of intellectual property (IP) cores, memory units, processors etc. [1]. In the state-of-the-art SoC designs, these blocks are connected by traditional busses to communicate and exchange data with each other. However, beyond a certain number of elements, bus-based systems encounter their communication limitations due to high power consumption, high bandwidth requirements and high latency [2]. Hence, the network-on-chip (NoC) paradigm has been proposed as an alternative solution to these communication bottlenecks. Bus-based systems enable communication among the resources via custom busses, shared busses, hierarchical busses or bus matrix as shown in Fig. 1.

In contrast, NoC-based systems include 2D, 3D mesh, Tree and Torus NoC designs. Also hybrid design of these architectures may be adopted [3, 4].

NoCs apply packet based switching for on chip communication. The packets enable the exchange of data among processing elements (PEs), using resource network interfaces (RNI), routers and interconnecting links as shown in Fig. 2. Routers forward packets across the network and may consist of input/output buffers, routing logic, allocators and crossbars [5]. Links/Channels enable the interconnection of the routers for data transmission. They are usually bidirectional and multi-layered. RNI is the interfacing unit between the PEs and the router. RNI and router collectively work as resource network router. Its function is to packetise and de-packetise the messages for processing and routing across the network. PE is the functional block that processes the data and may be an IP core, a memory element, a Processor etc.

The NoC architecture is specified by its topology, routing algorithm and flow control. Here, topology means the physical ordering and arrangement of the links connecting the network nodes. Exemplary topologies are Mesh, Torus, Folded Torus and Tree [6, 7]. The routing algorithm defines how messages between

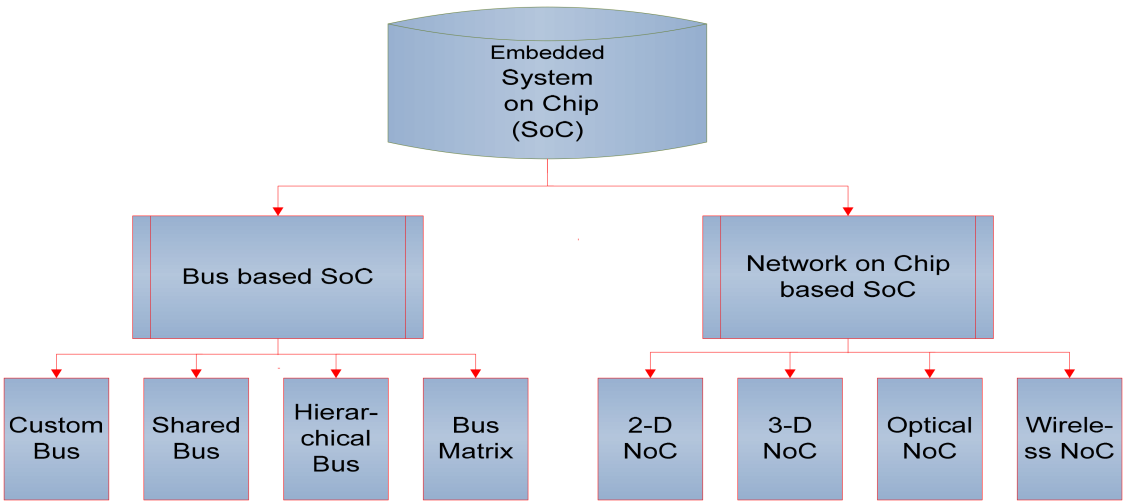

Fig. 1 *Embedded system categories*

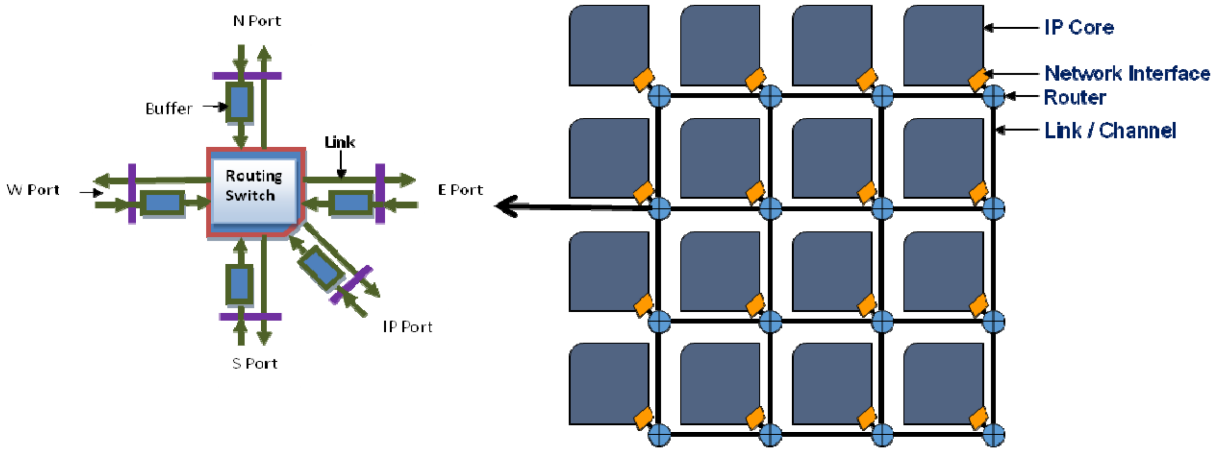

Fig. 2 *Elements of a common NoC*

PEs traverse the network. Routing may be deterministic or adaptive. A good routing algorithm will balance the traffic load across the network and should avoid deadlock and live-lock. Flow control is the allocation of communication resources for guaranteed and reliable transmission of packets between the PEs. In other words, flow control allocates the resources like buffers and link bandwidth to the messages as they traverse through the network [8–10].

NoCs are generally evaluated with respect to their performance parameters, such as energy consumption, area, communication bandwidth, throughput and latency [11]. Several simulators have been designed in order to evaluate these parameters and predict the system's performance prior to the design implementation [12–26]. Some simulators like ORION [12, 13] only focus on the performance of individual components, e.g. router power and area, while others such as Noxim, Nirgam and NoCTweak [14–17], are designed for measurement of the entire network performance. ORION simulator is used as a standard model for the estimation of energy of a router and link of the network. It is usually embedded in other network simulators (e.g. Noxim, Nirgam) to calculate the energy of the entire network. There are other energy models like CMOS standard library model [17] and Bit energy model [27] which are also used for estimation of the network energy consumption. As NoC inherits many features from general computer networks, so these network simulators can also be used for NoC system simulations [28–32]. Having in mind the rising amount of different simulators, this work is an attempt to provide a road-map that facilitates the selection of appropriate tools.

This paper is organised as follows: Section 2 presents related work on NoC simulators. Section 3 describes briefly the representation of NoC applications and Section 4 relates to the comparative analysis of NoC simulators. Simulation results are evaluated in Section 5 and concluding remarks are presented in Section 6.

2 Related work

Current research is mostly involved in topology design, router design, routing protocols, mapping and scheduling techniques [33]. Only limited study is available about NoC tools comparisons and surveys [34–42]. The work presented in [37] discusses some NoC proposals and contributions regarding their attainable capabilities. The author has collected a few NoC tools from the literature and presented a short description of the characteristics of the simulators. A comprehensive study is carried out about NoC concepts, but the survey lacks the comparative analysis of the NoC performance measurements.

In the research work of Neuenhahn [38], static and dynamic performance analysis of NoC is performed and evaluated at different abstraction levels. The static performance analysis has included timing models, while dynamic performance models are composed of FPGA, VHDL, Colored Petri Nets and SystemC-based simulations. Comparative analysis of different modelling techniques showed significant deviations in term of performance parameters of the simulators. The author has presented a quantitative analysis of the tools regarding their structure, but focused only on general aspects of the NoC tools.

The study in [39] presented an overview of the NoC concepts and simulation tools. Furthermore, the recent contributions on simulation tools are highlighted. The survey presented in [40] reviewed different approaches for NoC traffic models and performance evaluation. In this work, the evaluation-based analytical models and the design challenges of the NoC simulators are discussed, but only a descriptive study is presented about the NoC tools. The author of research work in [41] presented a short comparison of the HNOCS simulator with a few NoC simulators and highlighted the contribution of HNOC to the NoC research. The work presented in [42] collected numerous programming models and proposal for MPSoC but did not discuss dedicated NoC simulators. Availability of 3D, optical and wireless NoC simulators are also limited, because it is a new research area for the research community. It is an open problem for future NoC research to design specific simulators for NoC designs.

3 Representation of NoC applications

NoC systems are generally represented by characterisation graphs. Different simulators extract data from these graphs for performance simulations. An application can be represented by network task graph (NTG), which is subsequently scheduled on the available IPs through network core graph (NCG). NCG is then transformed and mapped on NoC packet switched network architecture through NoC architecture graph (NAG). In this work, these characterisation graphs are used to input data to the NoC simulators for comparative analysis and are briefly discussed in the following sections.

3.1 Network task graph

An NTG is a directed acyclic graph, $Gr = Gr(T, C)$ in which vertex of the graph characterises a task, ($t_i \in T$, $i = 1, 2, 3,\ldots$) for the computational resource of the application and represents information, such as execution time, energy consumption, task deadlines as shown in Fig. 3*a*. The directed arc ($c_{i,j} \in C$, $i = 1, 2, 3, \ldots, j = 1, 2, 3,\ldots$) represents either data or dependent information between processing tasks (t_i and t_j). The arc ($c_{i,j}$) is associated with a value (v ($c_{i,j}$)), which characterises the communicating data that are exchanged between the processing tasks. A real application is usually represented by NTG, which provides necessary information about the application for processing and simulations as shown in Fig. 3*b*.

3.2 Network core graph

An NCG, $Gr' = Gr'(P, A)$, is a directed graph in which vertex of the graph ($p_i \in P$) represent the PE, while the directed arc ($a_{i,j} \in A$) shows characteristic parameters between the PEs (p_i to p_j) as

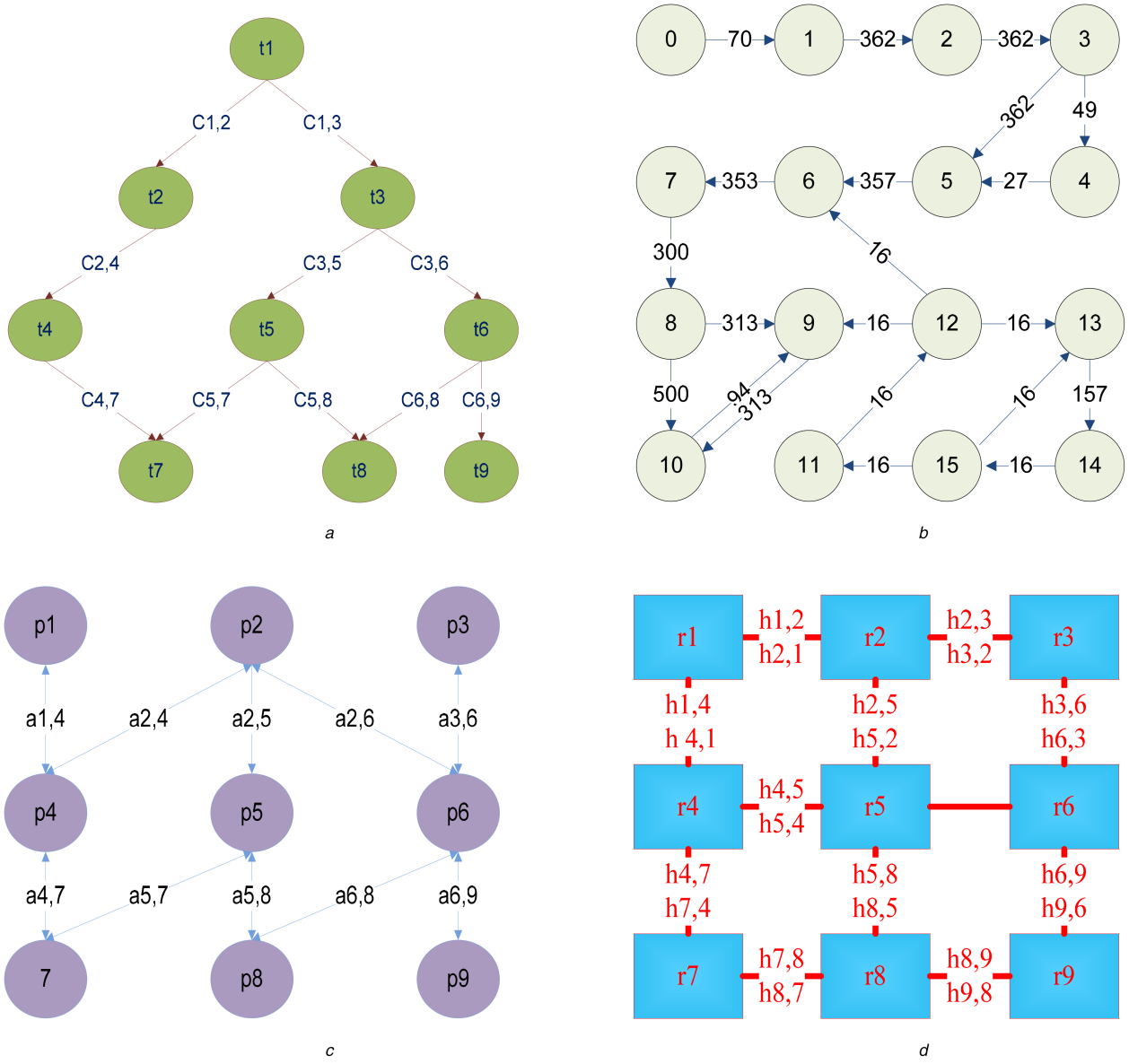

Fig. 3 *Representation of embedded applications*
(a) NTG, ***(b)*** VOPD application, ***(c)*** NCG, ***(d)*** NAG

shown in Fig. 3*c*. An arc ($a_{i,j}$) is associated with communication information, e.g. rate and volume, of communicating data and may have system design constraints, e.g. required latency and data bandwidth. NCG intrinsically represents the scheduling of the tasks (T) on the available PEs (P) for processing. When $P = T$, or $P > T$ then a single task can be scheduled on an individual PE, but when $P < T$, then two or more tasks should be scheduled on a single PE. For this purpose, a scheduler is required before performance simulations.

3.3 NoC architecture graph

NAG, $A = A(R, H)$, is an architecture graph, in which each vertex ($r_i \in R$) means a router (node) in the graph, while a directed arc ($h_{i,j} \in H$) indicates the bidirectional routing channel between the routers (r_i) and (r_j) as shown in Fig. 3*d*. Arc ($h_{i,j}$) is associated with $M_{i,j}$ (set of minimum paths from t_i to t_j) and $L(m_{i,j})$ that represents the set of all links associated with $m_{i,j}$. The cost of the arc is represented by $e(h_{i,j})$ which shows the average consumption of energy (joule) of sending data between network tiles (t_i and t_j), e.g. $E_{i,j}$ (bit). NAG represents the mapped applications on NoC tiles, which is then simulated on a specific tool for performance measurements.

To evaluate the potential of NoC simulators for mapping real application on NAG, video object plane decoder (VOPD) application is considered in this research work. The NTG of VOPD consists of 16 distinct tasks as shown in Fig. 3*b*. In this case, we assume that the tasks are equal to the available processors, therefore scheduling is not required. The NTG and NCG have one to one mapping, so we directly input the NTG to NoCTweak and NoCMAP for simulation as discussed in Section 5.2.

4 NoC simulators

A common challenge of selecting the right NoC simulator is that available tools usually are strong in certain measurements and having deficits in others. NoC simulator can be divided into two broad categories:

(1) General network simulators that can be used for NoC simulations (e.g. NS2, NS3, Omnet++, Wattch, Hotspot, Netsim, Gem5, Graphite, Hornet, Opnet, Fusionsim, Esece) [28–32].
(2) Specific NoC simulators, which are explicitly designed for NoC simulation (e.g. BookSim, HNOCS, WormSim, Ocsim, Vnoc, Matrics, SICOSY, Tpzsimul, Garnet, SUNMAP, Ocintsim, Noxim, Nostrum, Nirgam, Occn, Nocsim, NoCTweak, Atlas, Gpnocsim, Xmulator, NONMAP, ReliableNoC, MapoNoC, Phoenixsim, Access Noxim and ORION) [12–26, 43–48] (Table 1).

In this section, we describe some important simulators for NoC-based designs. Fig. 4 depicts the block diagram of a generic NoC simulator with its characteristic parameters. The NoC simulator may have the following input parameters:

(1) *Configuration options*: Configuration options define the type of application traffic simulated on the NoC tool. It may be synthetic traffic patterns or embedded application traces. It may also have a seed value selection for the simulation, log file for simulation outputs, warm-up time for the network to become stable and simulation run time selection.
(2) *Synthetic options*: Synthetic options define the size and type of topology for the traffic like 2D mesh and the type of synthetic traffic pattern such as random, transpose, bit-complement, bit-reverse, bit-shuffle, bit-rotate and hotspot routers selection. The hotspots are routers in the network that receive packetised data at a higher rate that they can handle. This phenomenon reduces system performance and can lead to deadlocks. An intelligent routing algorithm can prevent the formation of hotspots.
(3) *Embedded application traces*: Embedded applications are real application task graphs used in the simulation such as a VOPD, multimedia system, multi window display, MPEG4 decoder and E3S benchmarks.
(4) *Mapping option*: Mapping option such as near-optimal mapping (NMAP), simulating annealing (SA), branch and bound (BB) should also be included for obtaining optimised latency, throughput and energy consumption.
(5) *Traffic options*: Traffic options include the number of flits injected by each core per cycle (flit injection rate), the probability distribution of the period between two injected packets, packet length and flits per packet selection.
(6) *Router settings*: Router settings provide the type of router such as wormhole router, virtual channel router, shared queues router, bufferless router and circuit-switched router. It also defines pipeline type, the number of pipeline stages and buffer depth etc.
(7) *Routing options*: Routing options define routing algorithm like XY dimension-ordered routing, west-first, north-last and odd-even (OE) minimal adaptive routing. It may also have an output port selection such as the X dimension first, the dimension nearest to the destination first, the dimension farthest to the destination first, round-robin among output ports, the output port with highest credit first, switching arbitration policy and inter router link length.
(8) *Technology settings*: It includes CMOS technology process (e.g. 90, 65, 45, 32, 22 nm), clock frequency and supply voltage selection.
(9) *Measurement options*: The output measurements parameters include throughput, power, latency and energy consumption. The results of the output performance parameters predict the behaviour of the NoC multicore system before physical implementation.

4.1 NoCTweak

NoCTweak is a SystemC-based simulator developed by Tran and Baas [17] for NoC. It is designed for early exploration of performance, e.g. throughput, latency and energy estimation of NoC designs. NoCTweak uses standard CMOS library cell data for post layout timing and power estimation. The simulator has a command line interface for changing input simulation parameters. The current version of the simulator is designed only for 2D mesh topology, but has the option of synthetic and embedded traffic pattern. The hotspot and mapping options, i.e. random and NMAP algorithm for mapping the IPs are included. The router design selection, switching strategies, CMOS technology process selection, frequency and voltage selection are also included in this simulator.

4.2 Noxim

Noxim is developed by Maurizio Palesi, Davide Patti and Fabrizio Fazzino at the University of Catania [15, 16]. It is a SystemC-based simulator having a command line interface for changing input parameters. The simulator is based on ORION power model and currently designed for 2D mesh topology with only synthetic traffic pattern and wormhole router design. Energy, throughput and communication delay are its output statistic parameters. The input parameters include network size, packet size, packet injection rate, buffer depth, routing strategies and traffic distribution. The routing algorithms include XY, negative-first, west-first, north-last, OE, dyad, fully adaptive and lookup table based.

4.3 Nirgam

Nirgam is developed by Lavina Jain as a joint collaboration between the University of Southampton, UK and Malaviya National Institute of Technology, India [14]. It is an open source discrete event, cycle accurate simulator for NoC. The current version of the simulator is designed for 2D mesh and torus topologies. The wormhole switching mechanism is adopted in which packet consists of head, body and tail flits. Number of virtual channels, buffer size and clock frequency can also be changed for simulation. The simulator support source, XY and OE routing mechanism. Traffic option includes source/sink and synthetic generator which may be constant bit rate, bursty or input traced based. Performance parameters include average latency per packet (in clock cycles), average latency per flit (in clock cycles) and average throughput (in Gbps) for each channel.

4.4 Nostrum

Nostrum is a cycle accurate, layered NoC simulator written in SystemC and python by the Nostrum Team at Royal Institute of Technology (KTH), Stockholm [19]. It supports 2D mesh and torus topology with wormhole store and forward switching mechanism. The routing algorithms are XY and deflection order routing. It has the ability of mapping an application to the nodes of the network. The user can also change the arbitration policy and buffering options. It can be configured for both best effort and guaranteed communications. Best effort communication provides average performance, but better utilisation of network resources. Guaranteed communication is based on time division multiplexing with virtual circuits to insure guaranteed latency and throughput.

4.5 BookSim

BookSim is an interconnected network simulator written in C++ [18]. This simulator can simulate a wide range of topologies, e.g. 2D mesh, torus, concentrated mesh, fat tree, butterfly, flattened butterfly, quad tree and user specific any net etc. The simulator support input queued router and event-driven router micro-architecture with virtual channel support. The performance parameters are either latency or throughput versus offered load. The simulators also support changing buffer size, routing mechanism and arbitration policy. Currently, it only supports synthetic traffic patterns.

4.6 NOCMAP/reliableNoC

NOCMAP is an open source mapping simulator of NoC written in C++ by Hu *et al.* [27, 50]. Two mapping algorithms such as BB and SA are implemented in this tool. It uses the bit energy model to calculate the minimum total communication energy of the NoC. BB mapping algorithm is used for topological placement of IPs onto NoC platform to minimise total communication energy consumption. The link bandwidth is considered as constraint parameter. For comparison, ad-hoc SA method is also implemented, which indicates that BB is faster than SA technique with comparable results.

Based on the bit energy model EPAM (XY, OE and WF), which is energy and performance aware mapping with different routing algorithm (XY, OE and west-first), an efficient BB algorithm is implemented in this simulator. For comparison, SA algorithm is also implemented which shows that the proposed algorithms are more efficient than SA regarding result optimality and simulation speed. Moreover, it is observed that EPAM-OE gives more accurate results when applied to real and complex applications with large system size. ReliableNoC is an extended version of

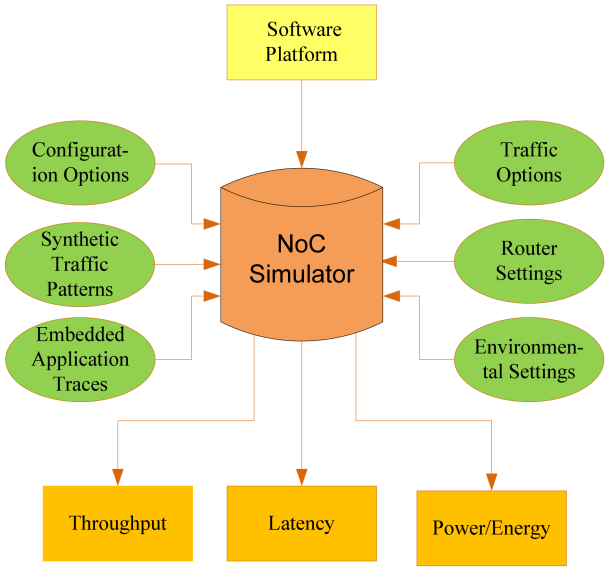

Fig. 4 *Generic NoC simulator*

NOCMAP simulator which has the addition of reliability parameter in NOCMAP simulator.

4.7 ORION

Architectural power estimation is an important factor in designing NoC-based systems. ORION is a fast and accurate architectural power model for router power and area simulation of NoC [12]. ORION 3.0 [13] is its latest release after ORION 1.0 and ORION 2.0. ORION 2.0 and ORION 3.0 have some additional functionality as well as closest performance parameters to the actual NoC designs. The comparison between ORION 2.0 simulations and Intel 80-core actual power consumption of Link, FIFO, Clock, Arbiter and Crossbar closely verify the simulation results of ORION 2.0 simulator.

4.8 Component-based interconnection network simulator (CINSim)

CINSim is a general purpose simulator for communication networks developed by a research group at Technische University Berlin (Real-Time Systems and Robotics) [28, 29]. The core of the simulator is written in C++ and has a command line interface. The editor of the simulator is written in JAVA and is platform independent. The graphical user interface (CINSim GUI) of the simulator is through an XML file of XML. The simulator has the capability of simulating both regular and irregular communication networks. The performance parameters are throughput, delay and latency. The network components are source buffers, non-shared buffers, routers, target buffers, routing and switching techniques and scheduling algorithms. Like NS-2 it can also be used for NoC simulations.

5 Comparative simulations

To carry out comparative analysis of NoC tools, a generic configuration setup is described in order to evaluate each simulator on the basis of its metrics of performance. The available simulators do not have a common approach for input/output parameter selection and also have different simulation models, so it is difficult to precisely scrutinise each simulator. Therefore a common configuration setup is selected in such a way that it has same input parameters for comparison. Parameters in Table 2 are used for analysis of the tools.

Most of the available simulators support 2D mesh, synthetic traffic pattern, wormhole switching mechanism, buffer depth availability and XY routing algorithm. Throughput, latency and power are the output characteristics parameters of these simulators.

In this research work, the results of different simulators are compared on the basis of a unified configuration setup for a 5×5, 2D mesh. The input parameters and the design structure of most of the simulators are different, so it is difficult to precisely compare all the simulators on a unified configuration setup. It is tried to compare some of these simulators on the basis of a best uniform approach. 2D mesh of size 5×5 is selected, because most of the

Table 1 Comparison of NoC simulators

Simulator	NoCTweak [17]	Noxim [15, 16]	Nirgam [14]	Nostrum [19]	BookSim [18]	WormSim [49]	NOCMAP [27, 50]	ORION [12, 13]
language	system C	system C	system C	system C	C++	C++	C++	C++
topology	2D mesh	2D mesh	2D mesh, torus	2D mesh, torus	wide range	2D mesh, torus	2D mesh	no
traffic pattern	synthetic, embedded	synthetic	synthetic, embedded	synthetic	synthetic	synthetic, embedded	synthetic, embedded	no
switching mechanism	wormhole virtual channel, Roshaq, bufferless, circuits switched	wormhole with virtual channel	wormhole with virtual channel,	wormhole with virtual channel	wormhole with virtual channel	wormhole with virtual channel	wormhole with virtual channel	user design
buffer depth option	yes	yes	yes	yes	yes	yes	yes	yes
routing algorithm	XY, negative first, west first, north last, OE, lookup table	XY, negative first, west first, north last, OE, dyad, fully adaptive, lookup table	source routing, XY, OE	XY, deflection routing	all	XY, OE, dyad	XY, OE, west first, dyad	no
performance parameters	power/energy consumption, throughput, latency	energy, throughput, communication delay	power, latency, throughput	latency, throughput, link utilisation	latency, throughput	energy	energy, reliability	router power, link power, router area
energy model	CMOS standard cell library model	ORION model	ORION model	no	no	ORION/bit energy model	bit energy model	ORION model
input parameters interface	command line	command line	log file	command line	log file	command line	command line	command line
hotspot option	yes	yes	no	no	no	yes	no	no
mapping	NMAP, random	no	manual	manual	no	no	BB, SA	no

Table 2 Platform description

Platform	Description	Remarks
topology	2D mesh	most widely used
network size	5 × 5 (25 nodes)	selected for simulation
workload/benchmark	synthetic	most simulators support
fixed packet length	10 flits	nominal size
flit injection rate	0.1–0.7 flits/cycle/node	range 0–1
traffic type	uniform	predictable results
router type	wormhole	most common
routing algorithm	XY	commonly supported
buffer size	8 flits	selected for simulation
input voltage	1 V	selected for simulation
operating clock frequency	1000 MHz	selected for simulation
warm-up time	20,000 cycles	for accuracy

simulator support 2D mesh topology and size 5×5 is chosen to minimise simulation time.

5.1 Synthetic benchmark

For comparison of NoC simulators, synthetic benchmark is selected because some simulator such as Noxim and CinSim support only synthetic traffic pattern. Injected traffic is also limited to 0.7 flits/cycle/node, in order to avoid network traffic congestion. In latency analysis, it is observed that Noxim and Nirgam have almost the same latency trend with injected traffic (flit injection rate) as shown in Fig. 5*a*. NoCTweak and CinSim have a non-linear response at low traffic, but when flit injection rate exceeds 0.5 flits/cycle/node; their response is comparable with Noxim and Nirgam. It may be due to the fact that NoCTweak and CinSim requires some extra parameters for analysis, which are limited in Noxim and Nirgam. Variations in the results are also due to the unavailability of an exact and uniform embedded platform in these simulators. The design structure of these simulators is also different from each other.

In throughput calculations (Fig. 5*b*), variation in results of NoCTweak is observed at the start, but the response is closely matched to that of Noxim and CinSim at a slightly high injection rate. Similarly, in Network energy analysis, NoCTweak and Noxim have a minor difference in results with the same injected traffic as shown in Fig. 5*c*. It is due to the fact that Noxim uses ORION energy model [13, 14] while NoCTweak utilises CMOS standard cell library model [17]. In network power calculations (Fig. 5*d*), again the results of NoCTweak and Nirgam are comparable and almost have the same trend at a high injection rate.

5.2 Real application benchmark

NoCTweak and NOCMAP have the ability to map a real application on NoC tiles for energy optimisation. A VOPD (Fig. 3*b*) is simulated on these simulators for energy comparison. VOPD has 16 tasks and therefore requires 4×4, 2D mesh for implementation. NoCTweak has built in random and NMAP mapping algorithms, while NOCMAP utilises BB and SA algorithms. In NoCTweak, the mapping through NMAP algorithm has 23.52% improvement in energy when compared with random algorithms as shown in Table 3.

Also, the BB algorithm in NOCMAP has 1% improvement in energy than SA algorithm. The simulation time of NMAP, SA, BB and Random algorithms of NoCTweak and NOCMAP simulator is also shown in Table 3. SA algorithm of NOCMAP is the slowest among all the other algorithms. As application mapping is non-polynomial hard problem, therefore, the SA algorithm in NOCMAP is not feasible for mapping of large applications.

A 24.89% improvement in power, 2% reduction in throughput and 22% improvement in latency of NMAP algorithm are also observed when compared with random mapping of VOPD application (Table 4). The mapping results of BB, SA, random and NMAP algorithms are shown in Fig. 6. As different algorithms

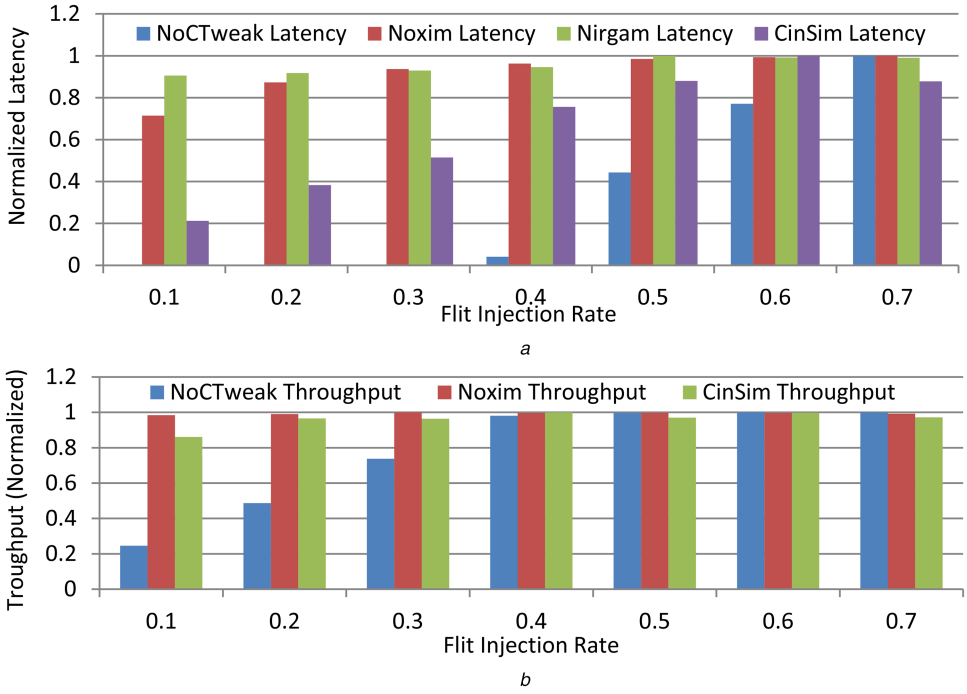

Fig. 5 *Continued*

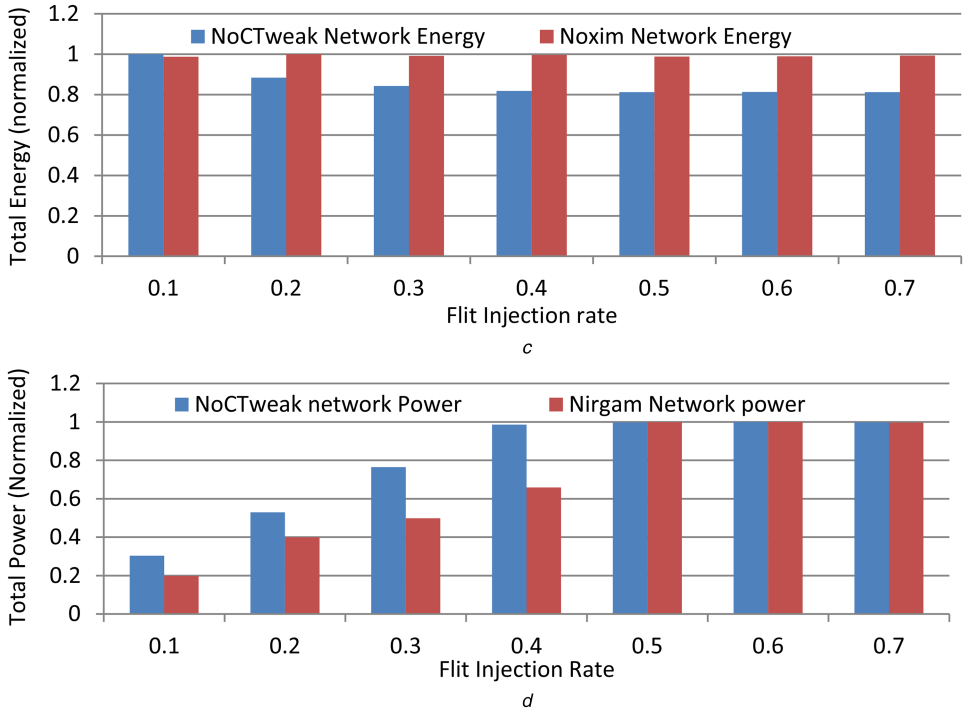

Fig. 5 *Performance parameters of NoC simulators*

(a) Network latency of NoCTweak, Noxim, Nirgam and CINSim, ***(b)*** Network throughput of NoCTweak, Noxim and CinSim, ***(c)*** Average energy of NoCTwek and Noxim, ***(d)*** Network power of NoCTweak and Nirgam

Table 3 VOPD application energy and simulation time comparison

	NoCTweak random mapping	NoCTweak NMAP mapping	NOCMAP BB mapping	NOCMAP SA mapping
energy consumption, pJ/flit	34.56	26.43	29.90	30.01
simulation time, s	14	14	16	150

Table 4 VOPD application performance comparison

Performance parameters	NoCTweak random mapping	NoCTweak NMAP mapping
average latency, cycles	21.81	17.01
average throughput, flits/cycle	0.048	0.047
total power, mW	26.30	19.76

generate different application mappings (Figs. 6*a–d*) due to the diverse design structure and convergence time, therefore the performance parameters differ from each other even under similar configurations. NMAP algorithm has the best application mapping for performance parameters among all the three algorithms. This efficiency of NMAP algorithm is due to the fact that it has better algorithm and design parameters for mapping the applications on network nodes.

The results and analysis of NoC tools divulge that an efficient mapping algorithm should be embedded in designing a complete NoC simulator for embedded system applications. Only synthetic traffic generators do not predict the real behaviour of NoC embedded systems. The mapping of the task on a specific network tile has a great impact on the performance parameters such as latency, power, throughput, and energy consumption of the network. Therefore real world applications should be properly scheduled and mapped on a specific topology before physical implementation.

6 Conclusion

This paper tries to pinpoint an important aspect of NoC research by analysing some renowned tools used for the simulation and comparison of NoC platforms. A comparative analysis of a few NoC open source simulators is also carried out to identify the strong and weak points of these simulators with respect to NoC performance parameters such as latency, throughput, power and energy consumption. Most of these simulators support 2D mesh topology, XY routing algorithm, wormhole switching mechanism and synthetic traffic pattern. The performance parameters such as energy/power, throughput and latency are compared with respect to the offered load (flit injection rate). Some NoC simulators also support area (ORION) and application mapping optimisation (NOCMAP, NoCTweak). The simulators and tools available for NoC are not mature, thereby creating a big challenge of standardisation to the research community in this important field of study. To accommodate most of the afore-mentioned performance parameters, generalised, accurate and precise simulators and tools are mandatory for modern embedded system designs. The future NoC research will mainly focus on optical, wireless and 3D NoC designs, therefore efficient simulators and tools will be the essential part of these designs.

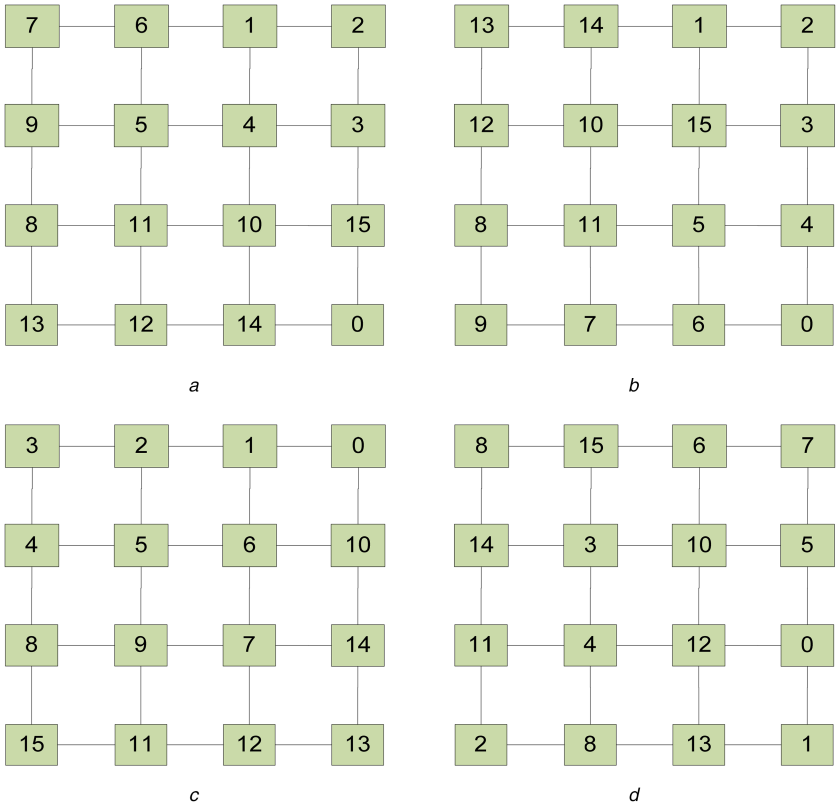

Fig. 6 *Comparison of VOPD application mapping of NOCMAP and NoCTweak*
(a) NOCMAP-BB mapping, ***(b)*** NOCMAP-SA mapping, ***(c)*** NoCTweak-NMAP mapping, ***(d)*** NoCTweak-random mapping

7 References

[1] Benini, L., De Micheli, G.: 'Networks on chips: a new SoC paradigm', *IEEE Comput. Soc.*, 2002, **35**, (1), pp. 70–78
[2] Dally, W.J., Towels, B.: 'Route packets, not wires, on-chip interconnection networks'. Proc. 38th DAC, 2001, pp. 684–689
[3] Choudhary, N.: 'Network-on-chip: a new SoC communication infrastructure paradigm', *Int. J. Soft Comput. Eng.*, 2012, **1**, (6), pp. 332–335
[4] Mineo, C., Davis, W.R.: 'The benefits of 3D networks-on-chip as shown with LDPC decoding'. IEEE Int. Conf. 3D System Integration, 2009. 3DIC 2009, pp. 1–8
[5] Jantch, A., Tenhunen, H.: '*Networks on chip*' (Kluwer Academic Publishers, 2003)
[6] Gulzari, U.A., Anjum, S., Torres, F.S., *et al.*: 'A new cross-by-pass-torus architecture based on CBP-mesh and torus interconnection for on-chip communication', *PLoS ONE* December 1, 2016, **11**, (12), pp. 1–18, e0167590
[7] Gulzari, U.A., Khan, S., Anjum, S., *et al.*: 'An efficient and scalable cross-by-pass-mesh architecture for on-chip communication', *IET Comput. Digit. Tech.*, 2017, (11), pp. 140–148
[8] Atienza, D., Angiolini, F., Murali, S., *et al.*: 'Network-on-chip design and synthesis outlook', *VLSI J.*, 2008, **41**, pp. 340–359
[9] Tsai, W.C., Lan, Y.C., Hu, Y.H., *et al.*: 'Networks on chips: structure and design methodologies', *Hindawi J. Electric. Comput. Eng.*, 2012, **2012**, Article ID 509465, 15 pages
[10] Sahu, S., Kittur, H.M.: 'Area and power efficient network-on-chip router architecture'. IEEE Conf. Information and Communication Technologies (ICT), 2013, pp. 855–859
[11] Gehlot, P., Singh Chouhan, S.: 'Performance evaluation of network-on-chip architectures'. Int. Conf. Emerging Trends in Electronic and Photonic Devices and Systems (ELECTRO-2009)
[12] Kahng, A.B.: 'ORION 2.0: a power-area simulator for interconnection networks', *IEEE Trans. VLSI Syst.*, 2012, **20**, (1), p. 191
[13] Kahng, A.B, Lin, B., Nath, S.: 'ORION 3.0: a comprehensive NoC router estimation tool', *IEEE Embedded Syst.*, 2015, **7**, (2), pp. 41–45
[14] Jain, L., Al-Hashimi, B., Gaur, M., *et al.*: 'NIRGAM: a simulator for NoC interconnect routing and application modeling'. Workshop on Diagnostic Services in Network-on-Chips, DATE, 2007, pp. 16–20
[15] Fazzino, F., Palesi, M., Patti, D.: 'Noxim: network-on-chip simulator', 2008
[16] Catania, V., Mineo, A., Palesi, M., *et al.*: 'Cycle-accurate network-on-chip simulation with Noxim', *ACM Trans. Model. Comput. Simul.*, 2016, **27**, (1), Article 4, 25 pages
[17] Tran, A.T., Baas, B.M.: 'NoCTweak: a highly parameterizable simulator for early exploration of performance and energy of networks on chip'. Technical Report, VLSI Computation Lab, ECE Department, UC Davis, July, 2012
[18] Jiang, N.: 'BookSim 2.0 user's guide', May 7, 2013
[19] Lu, Z.: 'NNSE: Nostrum network-on-chip simulation environment'. Swedish, System on Chip, 2005
[20] Murali, S., De Micheli, G.: 'SUNMAP: a tool for automatic topology selection and generation for NoCs'. Proc. 41st Design Automation Conf., 2004, pp. 914–919
[21] Jueping, C., Gang, H., Shaoli, W.: 'OPNEC-Sim: an efficient simulation tool for network-on-Chip communication and energy performance analysis'. 10th IEEE Int. Conf. Solid-State and Integrated Circuit Technology (ICSICT), 2010, pp. 1892–1894
[22] Hossain, H., Ahmed, M., Al-Nayeem, A., *et al.*: 'Gpnocsim – a general purpose simulator for network-on-chip'. Int. Conf. Information and Communication Technology, 2007. ICICT '07.Dhaka, 2008
[23] Lis, M., Shim, K., Cho, M., *et al.*: 'DARSIM: a parallel cycle-level NoC simulator'. 6th Annual Workshop on Modeling, Benchmarking and Simulation, Saint Malo, France, June 2010, pp. 1–10
[24] Chan, J., Parameswaran, S.: 'NoCGEN: a template based reuse methodology for networks on chip architecture'. IEEE 17th Int. Conf. VLSI Design, 2004, pp. 717–720
[25] Cong, J., Gururaj, K., Han, G., *et al.*: 'MC-Sim: an efficient simulation tool for MPSoC designs'. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD), 2008, pp. 364–371
[26] Lv, M., Guo, Y., Guan, N., *et al.*: 'RTNoc: a simulation tool for real-Time communication scheduling on networks-on-Chips'. Int. Conf. Computer Science and Software Engineering, 2008, vol. **4**, pp. 102–105
[27] Hu, J., Marculescu, R.: 'Energy-aware mapping for tile-based NoC architectures under performance constraints'. Asia and South Pacific Design Automation Conf., 2003, pp. 233–239
[28] Tutsch, D., Lüdtke, D., Walter, A., *et al.*: 'CINSim – a component based interconnection network simulator for modeling dynamic reconfiguration'. Proc. 12th Int. Conf. ASMTA, 2005, pp. 132–137
[29] Anjum, S., Munir, E.U.: 'Simulation and performance evaluation of network-on-chip architectures and algorithms using CINSIM', *J. Basic Appl. Sci. Res.*, 2011, **1**, (10), pp. 1594–1602
[30] Ning, W.: 'Simulation and performance analysis of network-on-chip architectures using OPNET', 1-4244-1132-7/07 © 2007 IEEE
[31] Ben-Itzhak, Y., Zahavi, E., Cidon, I., *et al.*: 'NoCs simulation framework for OMNeT++'. Fifth IEEE/ACM Int. Symp. Networks on Chip (NoCS), 2011, pp. 265–266
[32] Kourdy, R., Yazdanpanah, S., Rad, M.R.N.: 'Using the NS-2 network simulator for evaluating multi-Protocol label switching in network-on-Chip'. Second Int. Conf. Computer Research and Development, 2010, pp. 795–799
[33] Marculescu, R., Ogras, U.Y., Peh, L.S., *et al.*: 'Outstanding research problems in NoC design: systems, micro architecture, and circuit perspectives', *IEEE Trans. Computer-Aided Des. Integr. Circ. Syst.*, 2009, **28**, (1), pp. 03–21
[34] Agarwal, A., Iskander, C., Shankar, R.: 'Survey of network-on-chip (NoC) architectures & contributions', *J. Eng. Comput. Arch.*, 2009, **3**, (1), pp. 1–15
[35] Abbas, A.: 'A survey on energy-efficient methodologies and architectures of network-on-chip', *Comput. Electric. Eng. J.*, 2014, **40**, (8), pp. 333–347
[36] Kumar Sahu, P., Chattopadhyay, S.: 'A survey on application mapping strategies for network-on-chip design', *J. Syst. Archit.*, 2013, **59**, pp. 60–76

[37] Ben Achballah, A., Ben Saoud, S.: 'A survey of network-on-chip tools', *Int. J. Adv. Comput. Sci. Applic.*, 2013, **4**, (9), p. 61
[38] Neuenhahn, M.C., Schleifer, J., Blume, H.*, et al.*: 'Quantitative comparison of performance analysis techniques for modular and generic network-on-chip', *Adv. Radio Sci.*, 2009, **7**, pp. 107–112
[39] Alalaki, M.S, Agyeman, M.O: 'A study of recent contribution on simulation tools for network-on-chip', *Int. J. Comput. Electric. Autom. Control Inf. Eng.*, 2017, **11**, (4), pp. 33–37
[40] Qian, Z., Bogdan, P., Tsui, C.-Y.*, et al.*: 'Performance evaluation of NoC-based multicore systems: from traffic analysis to NoC latency modelling', *ACM Trans. Design Autom. Electron. Syst.*, 2016, **21**, (3), pp. 1–38
[41] Ben-Itzhak, Y., Zahavi, E., Cidon, I.*, et al.*: 'HNOCS: modular open-source simulator for heterogeneous NoCs'. Embedded Computer Systems (SAMOS), 2012 Int. Conf. Samos, 2013
[42] Fernandez-Alonso, E., Castells-Rufas, D., Joven, J.: 'Survey of NoC and programming models proposals for MPSoC', *IJCSI Int. J. Comput. Sci. Issues*, 2012, **9**, (2), No 3, pp. 22–32
[43] Amoretti, M.: 'Modeling and simulation of network-on-chip systems with DEVS and DEUS', *Hindawi Sci. World J.*, 2014, **2014**, Article ID 982569, pp. 1–9
[44] Dahir, N.S., Mak, T., Xia, F.*, et al.*: 'Modeling and tools for power supply variations analysis in networks-on-chip', *IEEE Trans. Comput.'*, 2014, **63**, (3), pp. 679–690
[45] Liu, W., Xu, J., Wu, X.*, et al.*: 'A NoC traffic suite based on real applications'. IEEE Computer Society Annual Symp. VLSI (ISVLSI), 2011, pp. 66–71
[46] Ghosh, D., Ghosal, P., Mohanty, S.P.: 'A highly parameterizable simulator for performance analysis of NoC architectures'. Int. Conf. Information Technology (ICIT), 2014, pp. 311–315
[47] Onizawa, N., Funazaki, T., Matsumoto, A.*, et al.*: 'Asynchronous network-on-chip simulation based on a delay-aware mode'. IEEE Computer Society Annual Symp. VLSI, 2010, pp. 357–362
[48] Genko, N., Atienza, D., De Micheli, G.*, et al.*: 'Feature – NoC emulation: a tool and design flow for MPSoC', *IEEE Circuits Syst. Mag.*, 2007, **7**, (4), pp. 42–51
[49] Ogras, U.Y., Marculescu, R.: 'It's a small world after all': NoC performance optimization via long-range link insertion', *IEEE Trans. Very Large Scale Integr. Syst.*, 2006, **14**, (7), pp. 693–706
[50] Hu, J.: 'Energy and performance aware mapping for regular NoC architectures', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 2005, **24**, (4), pp. 551–562